

UNITED STATES PATENT APPLICATION
FOR
METHOD AND LOGIC FOR
CAPTURING AND ANALYZING CONDUIT DATA

INVENTORS:

Colleen A. Barton

Daniel Moos

Prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CALIFORNIA 90025
(408) 720-8598

Attorney's Docket No. 003700.P002XC

"Express Mail" mailing label number: EV 410137622 US

Date of Deposit: 3-11-2004

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

Dawn R. Shaw

(Typed or printed name of person mailing paper or fee)

Dawn R. Shaw
(Signature of person mailing paper or fee)

3-11-2004
(Date signed)

METHOD AND LOGIC FOR CAPTURING AND ANALYZING CONDUIT DATA

RELATED APPLICATION

[0001] The present patent application is a continuation of U.S. Application Serial No. 09/657,759, filed September 8, 2000, entitled METHOD AND LOGIC FOR CAPTURING AND ANALYZING CONDUIT DATA, which is a continuation-in-part of U.S. Application Serial No. 09/340,216, filed June 25, 1999, now U.S. Patent No. 6,658,567 issued December 2, 2003, entitled METHOD AND LOGIC FOR LOCKING GEOLOGICAL DATA AND AN ANALYZER PROGRAM THAT ANALYZES THE GEOLOGICAL DATA .

[0002] The above-referenced applications are hereby incorporated herein by reference.

FIELD OF THE INVENTION

[0003] The present invention relates generally to the field of software and data distribution and, more specifically to the distribution of data and an associated analyzer program that analyzes the data to provide evaluations and assessments, the analyzer program being limited to analysis of the specific data.

BACKGROUND OF THE INVENTION

[0004] Pipelines are commonly used in the transportation of oil and gas. There are more than 300,000 miles of oil and gas pipelines in North America.

Construction costs are now of the order of \$1,000,000 per mile. The typical initial operating life of the pipelines are expected to be about 40 years, but 50% of the of the existing pipelines will be 40 years old at the year 2000.

[0005] Accurate monitoring of the pipelines is critical due to the potential risks to the environment when the pipelines rupture and due to the high costs of repair or replacement. Since oil and gas pipelines are normally buried, in-service inspection is performed by pumping a "smart electronic inspection pig" through the pipeline from one compressor station to the next.

[0006] Generally, the inspection tool detects and collects data indicating abnormalities (e.g., leakage, corrosion or metal loss) in the internal and external pipe surface or wall. The inspection tool may provide detailed signals about the condition of the pipelines. The signals are then converted to accurate estimates of defect size and geometry. This requires considerable expertise, as well as a detailed understanding of the effects of inspection conditions and the behavior of the type of pipeline steel used.

[0007] The information collected by the smart electronic inspection pig can be analyzed by an evaluation or analyzer software. The information can also be stored on a storage device such as, for example, a compact disc (CD) and can then be readily available for further analysis. The analyzer software typically reads large volumes of data generated during the inspection. The analyzer software may include a graphical user interface. Using the data collected during the inspection, the analyzer software may generally perform some data analysis and generates written and electronic reports or some form of graphical display.

[0008] Pipeline inspection activities or survey are generally performed by consulting firms such as pipeline assessment services performing both the data collection activity and the data analysis activity. An intelligent pig-based survey is expensive and may cost some hundreds of thousand of dollars, with certain long distance, more complicated lines being charged well in excess of this.

[0009] It would be cost advantageous if the user of the pipeline inspection data, (e.g., an oil company) could be enabled to have control of both the pipeline inspection data and the analyzer program that analyze the pipeline inspection data. This way the inspection data can be analyzed as often as desired and at any time as desired. However, the cost of an analyzer program may be prohibitively expensive and economically unattractive.

SUMMARY OF THE INVENTION

[00010] According to one embodiment of the invention, there is provided a method of locking a specific data and an analyzer program that analyzes the specific data. A first key is generated and associated with the specific data and a specific copy of the analyzer program. A gatekeeper logic is generated. The gatekeeper logic utilizes at least the first key to prevent the specific copy of the analyzer program from analyzing any other data except for the specific data.

[00011] According to another embodiment of the invention, there is provided a method of locking a specific conduit data with a specific copy of an analyzer program that analyzes the specific conduit data. A first key is generated and associated with both the specific conduit data and the specific copy of the analyzer program. A gatekeeper logic is generated. The gatekeeper logic utilizes at least the first key to prevent the specific copy of the analyzer program from analyzing any other conduit data except for the specific conduit data.

[00012] Other features of the present invention will be apparent from the accompanying drawings and from the detailed description which follows.

BRIEF DESCRIPTION OF THE DRAWINGS

[00013] The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

[00014] **Figure 1** is a block diagram illustrating an exemplary system for locking conduit data and an analyzer program that analyzes the conduit data.

[00015] **Figure 2** is a flow chart illustrating a method, according to an exemplary embodiment of the present invention, of locking conduit data and an analyzer program so that a specific copy of the analyzer program is enabled to analyze only specific conduit data.

[00016] **Figure 3A** is a flow chart illustrating a method, according to an exemplary embodiment of the present invention, of executing an analyzer program to analyze conduit data to which it is locked.

[00017] **Figure 3B** is a flow chart illustrating an alternative method, according to an exemplary embodiment of the present invention, of executing an analyzer program to analyze conduit data to which it is locked.

[00018] **Figures 4A – 4C** illustrate methods, according to alternative embodiments of the present invention, of distributing conduit data and an analyzer program that analyzes the conduit data to an end user of the conduit data conduit data conduit data and the analyzer program.

[00019] **Figure 5** is a flow chart illustrating a method, according to an exemplary embodiment of the present invention, that may be performed by a conduit assessment service provider, an end user, and an analyzer program software supplier to implement the method illustrated in **Figures 4A – 4C**.

[00020] **Figure 6** is a flow chart illustrating a method, according to an exemplary embodiment of the present invention, of distributing conduit data and an analyzer program to an end user.

[00021] **Figure 7** is a block diagram illustrating a machine, in the exemplary form of a computer system, within which a set of instructions for causing the computer system to perform any of the methodologies discussed above may be executed.

DETAILED DESCRIPTION

[00022] A method and logic for locking conduit data and an analyzer program that analyzes the conduit data are described. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be evident, however, to one skilled in the art that the present invention may be practiced without these specific details.

[00023] For the purposes of the present invention, the term “conduit” shall refer to any opening or cavity enclosed or surrounded by a structure having features, conditions or characteristics of a pipeline or pipeline-like environment such as, for example, an oil or gas pipeline, a sewer or other utility pipeline, a well or a bore hole. The structure may be generally vertical or horizontal or may form any angles with the ground level. The structure may be above ground, under ground, in land, under water, or any combination of these. Further, the term “data” shall be taken to include, but not limited to geomechanical, geophysical, in situ stress, petrophysical, geotechnical, acoustic wave form, magnetic wave resonance, vibration data, digital data or any other data derived using a logging device within the structure. The logging device may be a device that collects data such as, for example, a pig inspection device.

[00024] For the purposes of the present specification, the term “analyzer program” shall be taken to refer to any program that analyzes data for the purpose of presenting, interpreting or modifying the data. Accordingly, the operations performed by an analyzer program include, but are not limited

to, the extraction of data, the generating of data, the interpretation of data, the display of data, the filtering of data, and the enhancing of data.

[00025] Figure 1 is a block diagram illustrating a system 10, according to an exemplary embodiment of the present invention, for locking conduit data and an analyzer program that analyzes the conduit data. The system 10 includes conduit data files 12 that may, for example, comprise logged acoustic data, electrical data, optical data, digital data, magnetic data or vibration data. The conduit data files 12 are inputted to an encryption or locking routine 14 that operates to lock the conduit data files 12 to a specific copy of an analyzer program. This enables the specific copy of the analyzer program only to analyze the specific conduit data files 12, and no other conduit data files. To this end, the locking routine 14 includes a random number generator 16 that generates a first key 18, which in one embodiment of the present invention comprises a simple random number. In another embodiment, the first key 18 may comprise any random character sequence.

[00026] The locking routine 14 also operates to identify a specific characteristic of each of the conduit data files 12, and to assign a characteristic value to a characteristic parameter indicative of the particular characteristic of each conduit data file 12. For example, the characteristic parameter may be the size of a conduit data file 12, and the characteristic value attributed to this parameter may be actual size of a conduit data file 12 expressed in bits, bytes or any other measure.

[00027] Further, the locking routine 14 generates a second key 22 for each of the conduit data files 12, which is associated with a respective conduit data file 12, utilizing the first key and the characteristic value for a respective file 12. For example, the logging routine 14 may simply exclusive OR (XOR),

the first key and the characteristic parameter to generate the second key 22. The locking routine 14 then writes a respective second key 22 into a header portion of each of the files 12, as illustrated in **Figure 1**. Alternatively, the second key 22 may be inserted into the conduit data files 12 in a manner that is not so easily discernable. For example, the second key 22 may be fractured into pieces that are inserted at predetermined locations throughout the conduit data files 12 by the locking routine 14.

[00028] Further, the locking routine 14 generates a gatekeeper application 24, in the exemplary form of a Java™ applet, which implements an “unlocking” function with respect to the conduit data files 12 and a specific copy of an analyzer program. This is to permit the analyzer program to analyze the conduit data files. Further information regarding the gatekeeper application 24 is provided below.

[00029] A compiler 28 compiles the source code 26 of the analyzer program to generate a specific compiled copy of the analyzer application 30 that incorporates the first key 18 and the gatekeeper application 24. The source code 26 for the analyzer program can be provided by a software developer such as, for example, the GMI-Imager software developed by GeoMechanics International, Incorporated of Palo Alto, California.

[00030] The conduit data files 12 and the compiled analyzer application 30 may together comprise a single package 32 that is supplied to an end user. For example, the conduit data files 12 and the compiled copy of the analyzer application 30 may be written to a single compact disk (e.g., CD) that is supplied to the end user. Alternatively, the conduit data files 12 and the specific copy of the analyzer application 30 may be propagated to an end user from a source over a network.

[00031] In one embodiment, a single first key 18 may be utilized by the locking routine 14 to generate multiple distinct second keys 22 for each of the conduit data files 12. This may be achieved by performing an operation utilizing the first key 18 and a specific characteristic value for each distinct file of the conduit data files 12 to generate a distinct second key 22 for each such distinct conduit data file. Accordingly, the package 32 may comprise a single copy of the compiled analyzer application 30 that is able to analyze each of the multiple conduit data files 12, each having a different and unique second key 22.

[00032] **Figure 2** is a flow chart illustrating a method 34, according to an exemplary embodiment of the present invention, of locking conduit data and an analyzer program so that a specific copy of the analyzer program is enabled to analyze only the specific conduit data.

[00033] The method commences at block 36, with the generation of the first key 18 by the random number generator 16. At block 38, the locking routine 14 generates the gatekeeper application 24 that is to be compiled with the analyzer application source code 26 (e.g., the analyzer program). At block 40, the locking routine 14 determines a characteristic value for a characteristic parameter of at least one conduit data file 12. As discussed above, the locking routine 14 may determine a respective characteristic value for a number of conduit data files 12. The characteristic value may be, for example, the size of a respective conduit data file.

[00034] At block 42, the locking routine 14 then generates a respective second key 22 for each conduit data file 12 utilizing the first key 18 and the respective characteristic value 19 for the respective conduit data file 12. For example, the second key 22 may be generated for the respective conduit data

file 12 by performing an exclusive-or (XOR) operation between the first key 18 and the characteristic value 19 for each respective conduit data file 12.

[00035] At block 44, the locking routine 14 incorporates the second key 22 into the header of each respective conduit data file 12. As discussed above, in an alternative embodiment, a second key 22 for each respective conduit data file 12 may be distributed throughout the file 12 at predetermined and known locations. At block 46, the compiler 28 then compiles the analyzer application source code 26 together with the first key 18 and the gatekeeper application 24 and generates a specific compiled copy of the analyzer application 30. At block 48, the specific compiled copy of the analyzer application 30 and the conduit data files 12 that the analyzer application can access and analyze, are supplied to the end user as the package 32. The method 34 then ends at block 50.

[00036] **Figure 3A** is a flow chart showing a method 52, according to an exemplary embodiment of the present invention, of executing an analyzer program to analyze conduit data to which it is locked.

[00037] The method 52 commences at block 54, with the initiation or launch by an end user of the specific compiled copy of the analyzer application 30 shown in **Figure 1** and supplied to the user at block 48 of the method 34 illustrated in **Figure 2**. At block 56, the gatekeeper application 24 begins execution and it identifies the first key 18 within the specific copy of the analyzer application 30. At block 60, the gatekeeper application 24 opens the conduit data files 12 and, at block 62, determines a characteristic value for a characteristic parameter of each of the conduit data files 12. For example, the gatekeeper application 24 may ascertain the size of each of the conduit data files 12.

[00038] At block 64, the gatekeeper application 24 calculates a gate key for each of the conduit data files 12 utilizing the first key identified at block 58 and the respective characteristic value for characteristic parameter for each of the conduit data files 12. The calculation of the gate key is the same as the calculation of the second key 22 described in Figure 1 and at block 42 of Figure 2. For example, the gate key may be calculated by performing a XOR operation utilizing the first key 18 and the determined characteristic value for the characteristic parameter for each of the conduit data files 12.

[00039] At decision block 66, a determination is made as to whether the gate key generated for each of the conduit data files 12 corresponds to a respective second key 22 stored, for example, in the header portion of the respective conduit data file 12. Following a positive determination at decision box 66, for a specific conduit data file 12, the gatekeeper application 24 enables analysis of the specific conduit data file 12 by the specific copy of the analyzer application 30. On the other hand, following a negative determination for a specific conduit data file 12 at decision box 66, the gatekeeper application 24 disables the specific copy of the analyzer application 30 from analyzing the relevant data file 12. The negative determination occurs when the user attempt to use the specific copy of the analyzer application with a conduit data file other than the conduit data files 12. The method 52 then ends at block 72.

[00040] **Figure 3B** is a flow chart illustrating an alternative method 74, according to an exemplary embodiment of the present invention, of executing an analyzer program to analyze conduit data to which it is locked.

[00041] The method 74 corresponds substantially to the method 52 discussed above with respect to **Figure 3A**. They are different in that at

block 76, the gatekeeper application 24 identifies the second key 22 of a specific conduit data file 12, and then at block 78 calculates the gate key utilizing this second key 22 and the characteristic value of the relevant file 12. At decision box 80, the gatekeeper application 24 makes a determination as to whether the gate key corresponds to the first key 18 embedded within the specific copy of the analyzer application 30. This decision is to determine whether or not the specific copy of the analyzer application 30 will be enabled to analyze the relevant conduit data file 12.

[00042] In summary, the methods 52 and 74 differ in that, in the method 52, the first key 18 is utilized together with the characteristic parameter to determine the gate key which is then compared to the second key 22 stored within the conduit data file 12. In the method 74, the second key 22 is utilized to generate the gate key, that is then compared to the first key 18 embedded within the compiled and specific copy of the analyzer application 30.

[00043] **Figures 4A – 4C** illustrate methods 90, 92 and 94, according to alternative embodiments of the present invention, of distributing conduit data and an analyzer program that analyzes the conduit data to the end user of the conduit data and the analyzer program.

[00044] Referring first to **Figure 4A**, conduit data 98 may be provided by a conduit inspection service company 96 such as, for example, BlackHawk Pipeline Assessment Services of Atlanta, Georgia, to the end user 100. Conduit data 98 may be stored on a compact disk (CD) 99. The end user 100 then provides the conduit data 98 on the CD 99 to the analyzer software supplier 102.

[00045] Having received the conduit data 98 from the end user 100, the analyzer software supplier 102 will then proceed, utilizing the locking (or encryption) routine 14 to compile the analyzer application source code 26 to include the conduit data 98 and the locking (or encryption) routine 14 to thereby generate the package 32. As described above with reference to **Figure 1**, the package 32 comprises object code that constitutes a compiled specific copy of the analyzer application 30, and one or more conduit data files 12. The gatekeeper application 24, in conjunction with the first and second keys 18 and 22, constitutes a lock 104 illustrated in **Figure 4A**. The package 32 may be written to a compact disk 107, that is then supplied back to the end user 100.

[00046] Utilizing the compact disk 107, the end user 100 may then execute the analyzer application 30 to analyze only the conduit data 98. Except for the conduit data 98, the compiled analyzer application 30 will not work with any other conduit data.

[00047] The above-discussed method 90 of distribution of the locked compiled analyzer application 30 and conduit data 98 is advantageous to the analyzer software supplier 102. The analyzer software supplier 102 is able to incrementally recover the cost of the analyzer application by supplying multiple copies of an analyzer application 30 to the end user 100. Each analyzer application copy is locked to the specific conduit data 98. Each compiled and specific copy of the analyzer application 30 is supplied at a reduced price relative to the cost of supply of an unencumbered or “unlocked” analyzer program that is not limited to the specific conduit data 98, and that would be able to analyze any given conduit data. By supplying numerous copies of compiled specific analyzer applications 30 to the end

user, the analyzer software supplier 102 will thus be able to generate a steady revenue flow from the end user 100 and recover a "full" price for the program over time.

[00048] From the point of view of the end user 100, the distribution method 90 discussed above with reference to **Figure 4A** is advantageous. Instead of being required to buy an "unlocked" copy of the analyzer program 26 for a relatively high (and sometimes unaffordable) cost, the end user 100 incurs incremental and time-distributed costs for use of the analyzer program. A further benefit to the end user 100 is that the distribution method 90 implements an alternative to a "pay-per-use" system. This provides advantages in that the cost to the end user 100 of the analyzer application 30 is linked to the usage and value to the company of the specific conduit data 98.

[00049] **Figure 4B** shows an alternative method 92 of distributing conduit data and an analyzer program that analyzes the conduit data to an end user 100. The method 92 corresponds substantially to the method 90 with reference to **Figure 4A**. However, instead of the conduit data 98 being propagated between the conduit inspection service company 96, the end user 100 and the analyzer software supplier 102 on a compact disk, the relevant data and applications are transmitted via a network (not shown). The network may be a wire or wireless, and may comprise the Internet, a Wide Area Network (WAN) or a Local Area Network (LAN). The method 92 may provide some cost advantages over the method 90, and may also be more convenient in certain circumstances.

[00050] **Figure 4C** shows a further method 94 of distributing conduit data and an analyzer program, which are locked, to an end user 100. While the

conduit inspection service company 96 provides the conduit data 98 to the end user 100, as in the methods 90 and 92 above, this conduit data 98 is not provided to the analyzer software supplier 102. In the method 94, the analyzer software supplier 102 provides a further package 108 to the end user 100, the package 108 including a copy-protected and read-protected copy of the analyzer application source code 26, a copy of the locking routine 14, and a purge routine 106. The end user 100 then executes the locking routine 14 to enable the analyzer application 30 locally to generate the first key 18, the gatekeeper application 24 and the second key 22. Following compilation of a specific and compiled copy of the analyzer application 30, and the embedding of the second key 22 within a conduit data file 12, the purge routine 106 will then automatically be invoked to purge the locking routine 14 and the analyzer application source code 26 from a computer system operated by the end user 100. The end user 100 will then retain only the package 32, the other software having been purged from a relevant computer system by the purge routine 106.

[00051] The method 94 illustrated in **Figure 4C** is advantageous in that it is not required that the conduit data 98 be provided from the end user 100 to the analyzer software supplier 102. On the other hand, the generation of the locked analyzer application 30 and conduit data 98 at an end-user site may be undesirable, and may be unattractive to an end user 100.

[00052] **Figure 5** is a flow chart illustrating the steps performed by the conduit inspection service company 96, the end user 100 and the analyzer software supplier 102 as described above with reference to **Figure 4**. At block 110, the conduit data 98 is provided to the analyzer software supplier 102 from the conduit inspection service company 96 via the end user 100. At

block 112, the analyzer software supplier 102 then locks the analyzer application 30 to the conduit data 98. At block 114, the analyzer software supplier 102 supplies the locked analyzer application 30 and conduit data 98 to the end user 100. At block 116, the gatekeeper application 24, embedded within the analyzer application 30, allows a user to execute the locked software to analyze only the specific conduit data 98.

[00053] The method 90 described above with reference to **Figure 5** also embodies the processes performed when executing the method 92 discussed above with reference to **Figure 4B**.

[00054] **Figure 6** is a flow chart illustrating a method 94, according to an exemplary embodiment of the present invention, of distributing conduit data and an analyzer program to an end user. The method 94 is performed by the conduit inspection service company 96, the end user 100 and the analyzer software supplier 102.

[00055] At block 120, the end user 100 acquires the conduit data 98 from the conduit inspection service company 96. At block 112, the end user 100 requests an analyzer application source code 26, which is copy and read protected, to the end user 100. The supplier 102 also provides the locking routine 14 and purge routine 106. At block 126, the end user 100 then executes the locking routine 14 to lock the conduit data 98 to a specific and compiled copy of the analyzer application 30. This may involve generating the gatekeeper application 24, and performing a compile operation utilizing a compiler 28 as described with reference to **Figure 1**.

[00056] At block 128, the locking routine 14 calls the purge routine 106 to purge the locking routine and the analyzer application source code 26 from the computer system of the end-user 100. At block 130, the gatekeeper

application 24 allows the end user 100 to execute the analyzer application 30 to analyze the conduit data 98, and no other conduit data, to which is locked. The method 94 then ends at block 132.

[00057] **Figure 7** is a block diagram illustrating a machine, in the exemplary form of a computer system 140, within which a set of instructions, for causing the computer system 140 to perform any one of the methodologies discussed above, may be executed. The computer system 140 includes a processor 142, a main memory 144, and a static memory 146 that communicate with each other via a bus 148. The computer system 140 further includes a video display unit 149 (e.g., a liquid crystal display (LCD) or a cathode ray tube (CTR)). The computer system 140 further includes an alpha-numeric input device 150 (e.g., a keyboard), a cursor control device 152 (e.g., a mouse), a disk drive unit 154, a signal generation device 156 (e.g., a speaker) and a network interface device 158.

[00058] The disk drive unit 154 includes a machine-readable medium 160 on which is stored a set of instructions (i.e., software 162) embodying any one, or all, of the methodologies discussed above. The software 162 is also shown to reside, completely or at least partially, within the main memory 144 and/or within the processor 142. The software 162 may furthermore be transmitted or received via the network interface device 158.

[00059] For the purposes of this specification, the term " machine-readable medium" shall be taken to include any medium which is capable of storing or embodying a sequence of instructions for execution by the machine and that cause the machine to perform any one of the methodologies of the present invention. The term " machine-readable medium" shall accordingly

be taken to included, but not be limited to, solid-state memories, optical and magnetic disks, and carrier wave signals.

[00060] Thus, a method and logic for locking conduit data and an analyzer data program that analyzes the conduit data have been described. Although the present invention has been described with reference to specific exemplary embodiments, the present invention can also be practiced with any analyzer program being locked with any respective data to provide the end users the same advantages discussed above. For example, a sales person can purchase marketing data for a specific region locked together with an analyzer program that provides analyses of the same marketing data

[00061] It will be evident that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the invention. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.